

MicroPatent® PatSearch Fulltext: Record 1 of 1

Search scope: JP (bibliographic data only)

Years: 1836-2006

Patent/Publication No.: ((JP09146842))

Order/Download

Family Lookup

Find Similar

Legal Status

[Go to first matching text](#)

JP09146842 A
STORAGE SUBSYSTEM
 HITACHI LTD

Abstract:

PROBLEM TO BE SOLVED: To provide the storage subsystem with a controller which eliminates exclusive control over a cache between plural controllers sharing the cache. SOLUTION: The cache areas of caches 33 and 43 in which data are written mutually in multiple are divided by processors and the controllers 30 and 40 access only their controller control areas. The cache areas that the controllers use are fixed to eliminate the need for exclusive control between the processors and prevent deterioration in performance due to multiprocessor constitution.

[no drawing]

Inventor(s):

KOBAYASHI RIE
 MATSUMOTO YOSHIKO
 MURAOKA KENJI

Application No. 07300967 JP07300967 JP, **Filed** 19951120, **A1 Published** 19970606

Original IPC(1-7): G06F01208
 G06F01208 G06F00306 G06F00306

Current IPC-R	invention	version	additional	version
Advanced	G06F00306	20060101		
	G06F01208	20060101		
Core	G06F00306	20060101		
	G06F01208	20060101		

Patents Citing This One (1):

- US7032068 B2 20060418 NEC Corporation
Disk cache management method of disk array device

[Home](#)[Search](#)[List](#)[First](#)[Prev](#)

Go to

[Next](#)[Last](#)

For further information, please contact:

[Technical Support](#) | [Billing](#) | [Sales](#) | [General Information](#)

(11)特許出願公開番号

特開平9-146842

(43)公開日 平成9年(1997)6月6日

(51)Int.Cl. ⁶	識別記号	弁内整理番号	F I	技術表示箇所
G 0 6 F 12/08	3 2 0	7623-5B	G 0 6 F 12/08	3 2 0
		7623-5B		J
		7623-5B		H
3/06	3 0 2		3/06	3 0 2 A
	3 0 4			3 0 4 C

審査請求 未請求 請求項の数19 O L (全 14 頁)

(21)出願番号 特願平7-300967

(22)出願日 平成7年(1995)11月20日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 小林 利恵

神奈川県小田原市国府津2880番地 株式会社日立製作所ストレージシステム事業部内

(72)発明者 松本 佳子

神奈川県小田原市国府津2880番地 株式会社日立製作所ストレージシステム事業部内

(72)発明者 村岡 健司

神奈川県小田原市国府津2880番地 株式会社日立製作所ストレージシステム事業部内

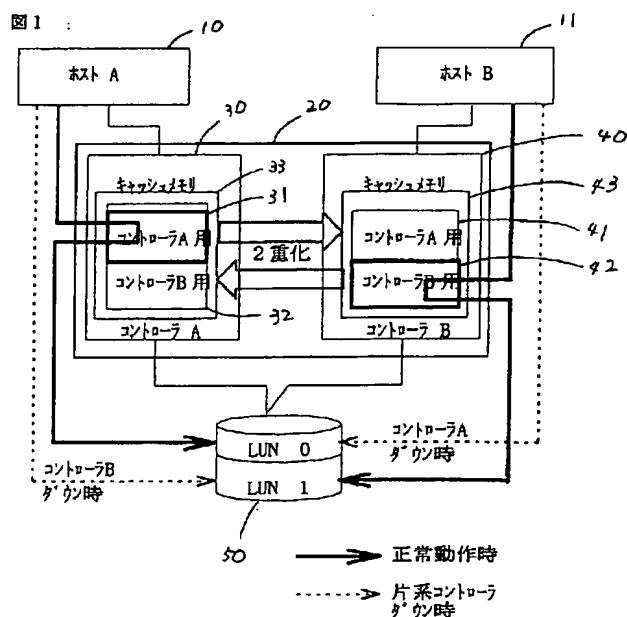
(74)代理人 弁理士 小川 勝男

(54) 【発明の名称】 記憶サブシステム

(57) 【要約】

【課題】キャッシュを共有する複数のコントローラ間のキャッシュの排他制御をなくした制御装置を有する記憶サブシステムを提供する。。

【解決手段】互いに多重書きされているキャッシュ 3 3、4 3において、キャッシュ領域を各プロセッサ毎に分割し、各々のコントローラ 3 0、4 0は、自コントローラ制御エリアのみにアクセスする。各コントローラが使用するキャッシュ領域を固定化することにより、プロセッサ間の排他制御を不要とし、複数プロセッサ化に伴う性能劣化を防止することが可能となる。



【特許請求の範囲】

【請求項 1】 ホストコンピュータのデータを格納し、複数の記憶領域を有する記憶装置と、
該ホストコンピュータの指示に基づいて該記憶装置の制御を行い、該ホストコンピュータと該ディスク装置との間のデータ転送を制御し、該ホストコンピュータと該記憶装置との間を転送されるデータを一時的に保持する複数の領域を有するキャッシュメモリを有する複数のコントローラと前記複数のコントローラ間を接続するバスとを具備する制御装置とを有する記憶サブシステムであって、
前記コントローラには、前記記憶装置の複数の記憶領域のうち少なくとも 1 つと該コントローラのキャッシュメモリの複数の領域のうち少なくとも 1 つと前記バスにより接続される他のコントローラのキャッシュメモリの複数の領域のうち少なくとも一つが割当てられることを特徴とする記憶サブシステム。

【請求項 2】 請求項 1 記載の記憶サブシステムにおいて、前記コントローラは、前記ホストコンピュータから転送されるデータを該コントローラに割当てられている複数の前記キャッシュメモリに書込むことを特徴とする記憶サブシステム。

【請求項 3】 請求項 2 記載の記憶サブシステムにおいて、前記コントローラに障害が発生したときは、前記他のコントローラは該障害コントローラが担当していた前記記憶装置の記憶領域の処理を行うことを特徴とする記憶サブシステム。

【請求項 4】 請求項 3 記載の記憶サブシステムにおいて、該他のコントローラはホットスタンバイしているコントローラであって、ホットスタンバイしているコントローラには、キャッシュメモリの記憶領域を割り当てないことを特徴とする記憶サブシステム。

【請求項 5】 請求項 1 記載の記憶サブシステムにおいて、前記制御装置は複数の前記コントローラ間を接続するバスを有し、前記コントローラが他のコントローラに割当てられた前記記憶装置の記憶領域に対する処理要求をホストコンピュータから受取ったときは、前記コントローラは、前記他のコントローラに該処理要求を通信することを特徴とする記憶サブシステム。

【請求項 6】 請求項 1 記載の記憶サブシステムにおいて、前記キャッシュ領域の分割は、コントローラの負荷に応じて変更することを特徴とする記憶サブシステム。

【請求項 7】 ホストコンピュータのデータを格納する複数の論理ボリュームを有する磁気ディスクと、
該ホストコンピュータと該ディスク装置との間を転送されるデータを一時的に保持する複数の領域を有するキャッシュメモリと、前記キャッシュメモリとが接続され、該データのデータ転送を制御するデータ転送制御部とを有する複数のコントローラと、複数のコントローラ間を接続するバスとを有し、該ホストコンピュータの指示に

基づいて該磁気ディスク装置の制御を行う制御装置とを有する記憶サブシステムであって、

前記コントローラには、前記磁気ディスク装置の複数の論理ボリュームのうち少なくとも 1 つと該コントローラのキャッシュメモリの複数の領域のうち少なくとも 1 つと、他のコントローラのキャッシュメモリの複数の領域のうち少なくとも 1 つとが割当てられることを特徴とする記憶サブシステム。

【請求項 8】 請求項 7 記載の記憶サブシステムにおいて、前記コントローラは、前記ホストコンピュータから転送されるデータを該コントローラに割当てられている該コントローラのキャッシュメモリの領域と、該コントローラに割当てられている他のコントローラのキャッシュメモリの領域とに書込むことを特徴とする記憶サブシステム。

【請求項 9】 請求項 8 記載の記憶サブシステムにおいて、前記コントローラに障害が発生したときは、前記他のコントローラは該障害コントローラが担当していた前記論理ボリュームの処理を行うことを特徴とする記憶サブシステム。

【請求項 10】 請求項 9 記載の記憶サブシステムにおいて、前記他のコントローラはホットスタンバイしているコントローラであって、ホットスタンバイしているコントローラには、キャッシュメモリの記憶領域を割り当てないことを特徴とする記憶サブシステム。

【請求項 11】 請求項 7 記載の記憶サブシステムにおいて、前記コントローラが他のコントローラに割当てられた論理ボリュームに対する処理要求をホストコンピュータから受けとったときは、前記コントローラのデータ転送制御部は、該他のコントローラに前記第一のバスを介して処理要求を転送し、該処理要求を受領した該他のコントローラが該論理ボリュームに対する処理を行い、処理結果を、前記コントローラに転送することを特徴する記憶サブシステム。

【請求項 12】 請求項 7 記載の記憶サブシステムにおいて、前記キャッシュ領域の分割は、コントローラの負荷に応じて変更することを特徴とする記憶サブシステム。

【請求項 13】 請求項 7 記載の記憶サブシステムにおいて、前記コントローラ間のバスは、2 つの前記コントローラを接続する第一のバスと、該 2 つのコントローラの組を接続する第二のバスを含むことを特徴とする記憶サブシステム。

【請求項 14】 請求項 13 記載の記憶サブシステムにおいて、前記制御装置にコントローラを増設するときは、前記コントローラの 2 台単位に増設することを特徴とする記憶サブシステム。

【請求項 15】 ホストコンピュータのデータを格納し、複数の記憶領域を有する記憶装置と、
該ホストコンピュータの指示に基づいて該記憶装置の制御を行い、該ホストコンピュータと該記憶装置との間の

データ転送を制御し、該ホストコンピュータと該記憶装置との間を転送されるデータを一時的に保持する複数の領域を有するキャッシュメモリを有する複数のコントローラと前記複数のコントローラ間を接続するパスとを具備する制御装置とを有する記憶サブシステムであって、前記コントローラには、前記記憶装置の複数の記憶領域のうち少なくとも1つと該コントローラのキャッシュメモリの複数の領域のうち少なくとも1つと前記パスにより接続される他のコントローラのキャッシュメモリの複数の領域のうち少なくとも一つが割当てられ、前記ホストコンピュータから転送されるデータは、該コントローラに割当てられている該コントローラのキャッシュメモリの領域と、該コントローラに割当てられている他のコントローラのキャッシュメモリの領域に書込まれることを特徴とする記憶サブシステム。

【請求項16】請求項15記載の記憶サブシステムにおいて、前記コントローラに障害が発生したときは、前記他のコントローラは該障害コントローラが担当していた前記記憶装置の記憶領域の処理を行うことを特徴とする記憶サブシステム。

【請求項17】請求項15記載の記憶サブシステムにおいて、該他のコントローラはホットスタンバイしているコントローラであって、ホットスタンバイしているコントローラには、キャッシュメモリの記憶領域を割り当てないことを特徴とする記憶サブシステム。

【請求項18】請求項15記載の記憶サブシステムにおいて、前記制御装置は複数の前記コントローラ間を接続するパスを有し、前記コントローラが他のコントローラに割当てられた前記記憶装置の記憶領域に対する処理要求をホストコンピュータから受取ったときは、前記コントローラは、前記他のコントローラに該処理要求を通信することを特徴とする記憶サブシステム。

【請求項19】請求項15記載の記憶サブシステムにおいて、前記キャッシュ領域の分割は、コントローラの負荷に応じて変更することを特徴とする記憶サブシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、上位装置からの情報の入出力要求を制御する制御装置を有する記憶サブシステムに関し、特に、制御装置内のコントローラ及びキャッシュメモリを冗長構成とする記憶サブシステムに関する。

【0002】

【従来の技術】コントローラ及びディスク等の記憶装置に冗長性を持たせた記憶サブシステムとして、一方の系が現用系として、他方の系が予備系として稼働する2重の系で構成される記憶サブシステムがある。

【0003】特開平4-215142に記載されている記憶サブシステムは、現用系のディスク装置の記憶情報

を両系からアクセス可能な共用ディスク装置を介して予備系のディスク装置に複製すること、あるいは、現用系コントローラ障害時は、予備系のコントローラによって、現用系のディスク装置の記憶情報を抽出可能とすることによって、コントローラ及びディスク装置障害時のデータ保全性の向上を計っている。

【0004】

【発明が解決しようとする課題】最近の市場動向として、記憶装置の高性能化、大容量化、低価格化の要求が高まっており、RAIDの技術が重視されている。RAIDの技術を適用した記憶サブシステムにおいては、複数のディスク装置をアレイ状に構成する。そして、データ書き込み時には、書き込みデータに加えて冗長データを書き込みデータを格納したディスク装置とは異なるディスク装置へ書き込む。アレイ構成内の任意のディスク装置の故障に対しは、他のディスク装置のデータと前記冗長データから障害ディスク装置上のデータを修復可能とすることによって、ディスク装置のデータの保全性の向上を計っている。

【0005】しかし、RAIDの技術を適用した記憶サブシステムは、データの保全性が向上する反面、上述した冗長データ生成／書き込みのために処理時間が増大するため、ホストからのI/O処理と同期して冗長データの生成／書き込みまでを行うと、ホストからのライト性能が大幅に劣化する。従って、RAIDの技術を適用した記憶サブシステムのコントローラには、ライトキャッシュが不可欠となっている。

【0006】ライトキャッシュとは、コントローラ内に搭載された、データを一時的に書き込むキャッシュであり、ホストからのライト要求では、このキャッシュに書き込みを行った時点で、ホストに終了報告を返す。そして、ホストのI/O処理とは非同期に冗長データの生成、ライトデータ及び冗長データのディスク装置への格納を行うことにより書き込み処理時の性能低下を防ぐ。しかし、ライトキャッシュを用いると、データをキャッシュ上に書き込んだ時点でホストに終了報告をするため、キャッシュ上にディスク装置未反映のホストデータが存在する。従って、キャッシュに冗長性がなければ、キャッシュ障害時に、ユーザデータロスとなる。したがって、特にデータの信頼性が強く求められる記憶サブシステムに用いる制御装置では、従来のコントローラ、記憶装置の冗長構成に加え、一般的にキャッシュにも冗長性を持たせることが行われている。

【0007】コントローラを多重化した記憶サブシステムにおいて、単にキャッシュを多重化すると、キャッシュ上のデータを複数の制御装置から同時にアクセスすることによるデータ整合性矛盾を防ぐためにキャッシュアクセス時に複数の制御装置からの排他制御が必要になる。そして、コントローラを多重化した記憶サブシステムでは、この排他制御により、シングルコントローラの

記憶サブシステムに比べて性能が低下する。

【0008】本発明の目的は、コントローラの多重化及びキャッシュの多重化に伴う、プロセッサ間のキャッシュの排他制御を無くし、性能を落とすことなく信頼性を上げることにある。

【0009】

【課題を解決するための手段】上記の目的を達成するため、本発明による記憶サブシステムは、各プロセッサ毎に処理担当の論理ボリュームを排他的に決める手段と、あるプロセッサが受領したホストコンピュータからの要求が、担当外であった場合は、担当プロセッサに処理要求を通信する手段と、上記通信を受領したプロセッサは、処理結果を要求元プロセッサに通信する手段と、各プロセッサ毎に、ディレクトリ／データセグメント等のキャッシュ構成要素を持つ手段と、上記構成要素の状態をプロセッサの負荷に応じてダイナミックに変更する手段と、ホストコンピュータからのライトデータを複数のコントローラ上のキャッシュへ多重書きする手段と、コントローラ障害時には、障害コントローラ内プロセッサの持つキャッシュ構成要素の制御権を正常系コントローラ内のプロセッサに切り替える手段と、コントローラ復旧時には、該制御権を復旧プロセッサに戻す手段と、ディスク装置への書き込み時にキャッシュメモリ障害が発生した際は、多重書きしている他キャッシュからディスク装置に書き込みを行う手段とを有する。

【0010】上述した手段により、複数のプロセッサ間で、キャッシュを排他制御することなく複数のコントローラ上のキャッシュへ多重書きを行うことができ、複数プロセッサ化に伴う性能低下の発生を防ぎ、性能を落とすことなく信頼性の向上を計ることができる。

【0011】また、上述の手段により、キャッシュメモリ障害時には、多重書きしている他キャッシュからディスク装置への書き込みを行い、データロストを防止できる。

【0012】さらに、上記手段により、コントローラ障害時には、自動的に、正常系に切り替えて処理続行が可能であり、また、コントローラ復旧時には、自動的に、復旧系に処理を戻すことが可能となり、システムの無停止運用を実現できる。

【0013】

【発明の実施の形態】図1は、本発明の概念図である。

【0014】図1において、10、11はホストコンピュータ、20はデュアルコントローラ構成をとる制御装置、50はディスク装置であり、ディスク装置50は、論理ボリューム0と論理ボリューム1の2つの論理ボリュームに分割されている。

【0015】ホストA10は、制御装置20内のコントローラA30を介して、論理ボリューム0の処理を行っており、ホストB11は、制御装置20内のコントローラB40を介して、論理ボリューム1の処理を行って

る。

【0016】ここで、コントローラA30には論理ボリューム0が、コントローラB40には論理ボリューム1が処理担当論理ボリュームとして割当てられている。又、コントローラ内のキャッシュの領域は、それぞれ、コントローラA用キャッシュ31、41、コントローラB用キャッシュ32、42に2分割されている。そして、コントローラA用キャッシュ31と41の間で2重書きを行い、又、コントローラB用キャッシュ32と42の間でも2重書きを行う。

【0017】コントローラA30は、通常、コントローラA用キャッシュ31と41を用いて、I/O処理を行い、同様に、コントローラB40は、コントローラB用キャッシュ32と42を用いて、I/O処理を行う。このように、コントローラ毎に使用するキャッシュ領域を個別に割り当てることにより、コントローラ間の排他制御を無くし、コントローラ数増加に伴う性能劣化を防ぐことができる。

【0018】また、コントローラB40障害時には、コントローラB用キャッシュ32、42をコントローラA30が使用することにより、ホストA10からコントローラA30を介して、コントローラB40の処理担当であった論理ボリューム1への処理を続行させることができる。

【0019】以下、本発明によるマルチコントローラ構成の制御装置の1実施例を図面を用いて説明する。

【0020】図2は、本発明をマルチコントローラ構成の磁気ディスクアレイサブシステムに適用した場合の構成図である。

【0021】図2において、1000、1100、1200、1300はデータ処理を行う中央処理装置であるホストコンピュータ、2000はマルチコントローラ構成をとりディスク装置の制御を行う制御装置、7000、7100はホストコンピュータのデータを格納するディスク装置である。ここで、制御装置2000は、ホストバスに直結したスロットに差し込みホスト筐体内に組み込む場合もあるし、制御装置として独立した筐体に組み込む場合もあるし、ディスク装置を組み込んだ筐体として実現する場合もある。また、ディスク装置群7000及び7100は、データディスクとパリティディスクからなるパリティグループを含んでいる。さらに、ディスク装置群7000は、論理ボリューム0と論理ボリューム1とに、ディスク装置群7100は論理ボリューム2と論理ボリューム3とに分割されている。

【0022】制御装置2000は、ホストコンピュータ1000、1100とディスク装置7000間のデータ転送を制御するコントローラ3000、4000及びホストコンピュータ1200、1300とディスク装置7100間のデータ転送を制御するコントローラ5000、6000より構成される。

【0023】コントローラ3000は、ホストコンピュータ1000とのプロトコル制御を行うホストI/F制御部3100、コントローラ全体を制御するマイクロプロセッサ（以下「プロセッサ」という。）3200、データの転送を実行するデータ転送制御部3300、ホストコンピュータ1000とディスク装置7000のデータ転送時及びプロセッサ間通信時に用いられるキャッシュ3400、各ディスク装置7000とのプロトコル制御を行うDRVI/F制御部3500より構成される。コントローラ4000、5000、6000はコントローラ3000と同一の構成である。

【0024】プロセッサ3200は、後述の手段により、あらかじめプロセッサ毎に排他的に割り当てた担当論理ボリュームの処理を行う。このプロセッサ毎の担当論理ボリュームの指定は、ホストコンピュータから論理ボリューム毎の担当プロセッサ指定コマンドを受け取ることであり、ダイナミックに設定可能である。このプロセッサと担当論理ボリュームとの対応情報は、後述のキャッシュ上の共通メモリ領域3410、4410に格納する。

【0025】データ転送制御部3300はプロセッサ3200からの指示により、ホストコンピュータ1000からのライトデータを指定キャッシュに多重書きする機能を備えている。この実施例の構成では、キャッシュ3400とキャッシュ4400の間で2重書きを行い、また、キャッシュ5400とキャッシュ6400の間でも2重書きを行う。以下、キャッシュ3400とキャッシュ4400の2面に2重書きする方式について説明する。

【0026】キャッシュ3400とキャッシュ4400の内容について図3を用いて説明する。尚キャッシュ3400とキャッシュ4400は内部構成が同一であるため、キャッシュ3400を例に説明する。キャッシュ3400は、プロセッサ間通信に用いる制御情報を格納している共通メモリ領域3410、プロセッサ3200用領域3480、プロセッサ4200用領域3490より構成される。

【0027】プロセッサ3200用領域3480は、ホストコンピュータとディスク装置間のデータ転送時、データを1次的に格納するデータ格納エリア3482、データ格納エリア3482を管理するデータ管理情報3481より構成され、データ格納エリア3482に格納するライトデータと、このライトデータの管理情報は、キャッシュ4400内のプロセッサ3200用領域4480に2重書きを行う。同様に、プロセッサ4200用領域3490は、プロセッサ4200により、キャッシュ4400内のプロセッサ4200用領域4490のライトデータとライトデータの管理情報が2重書きされている。

【0028】共通メモリ領域3410は、論理ボリューム

ム担当プロセッサ情報3420、プロセッサ負荷情報3430、多重書き情報3450、プロセッサ間コミュニケーションメモリ3460より構成され、これらの情報は全て、データ転送制御部3300、4300によって、キャッシュ3400と4400に2重書きされている。

【0029】図3(c)にプロセッサ間コミュニケーションメモリの構成を示す。プロセッサ間コミュニケーションメモリ3460は、プロセッサ3200、4200、5200、6200毎の書き込み用メモリ3461、3462、3463、3464より構成される。図3(d)にプロセッサ書き込み用メモリの構成を示す。プロセッサ3200書き込み用メモリ3461は、自プロセッサ以外のプロセッサ4200、5200、6200への要求用エリア3471、3472、3473と自プロセッサ以外のプロセッサ4200、5200、6200からの要求に対する応答用エリア3474、3475、3476より構成される。プロセッサ4200、5200、6200書き込み用メモリ3462、3463、3464の内部構成は、プロセッサ3200書き込み用メモリ3461と同一構成である。

【0030】キャッシュ5400とキャッシュ6400との間も、共通メモリ領域を除いて、キャッシュ3400とキャッシュ4400との間と同様に2重化が行われている。共通メモリ領域は、キャッシュ3400、4400に2重書きされている情報を制御装置内の全プロセッサで共有するため、キャッシュ5400、6400には存在しない。

【0031】本発明を実施する制御装置では、コントローラの増設はコントローラ2台単位で行い、対になったコントローラのキャッシュ間のみで2重書きを行うとともに、ドライブ側のデータバスについても、それぞれのディスク装置は対になったコントローラにのみ接続することによりハードウェア構成を簡略化し、ドライブ側データバス上の競合を回避することが可能となる。

【0032】次に本実施例における、磁気ディスクサブシステムでの、ホストコンピュータ1000からのI/O処理について図4、図5、図6を用いて説明する。まず最初に、プロセッサ3200担当論理ボリュームへのI/O処理について説明する。

【0033】図4は、ホストからのI/O処理を示すフローチャートである。ホストコンピュータ1000からの書き込み要求時、プロセッサ3200は、まず、共通メモリ領域3410内の論理ボリューム担当プロセッサ情報3420によって、処理要求論理ボリュームの担当プロセッサ情報を取得し、自処理担当論理ボリューム(LUN)への処理かの判定を行い(ステップ902)、自プロセッサ処理担当論理ボリュームへの処理であることを認識する。次に、処理種類の判定を行い(ステップ903)、書き込み処理であることを認識する。

ホストI/F制御部3100により、書き込み論理データを受領し、データ転送制御部3300によってキャッシュ3400のコントローラ3000用領域3480とキャッシュ4400のコントローラ3000用領域4480とにその管理情報とともに2重に格納する(ステップ904)。そして、この時点でホストコンピュータ1000に終了を報告する(ステップ905)。

【0034】図5は、キャッシュ内のデータをディスク装置に格納する処理を示すフローチャートである。プロセッサ3200は、ホストコンピュータ1000からのI/O処理とは非同期にプロセッサ3200用領域3480上のライトデータをデータ転送制御部3300とDRV I/F制御部3500によりディスク装置群7000に格納する(ステップ922)。この際、キャッシュのメモリ障害により読み込みエラーが発生した場合(ステップ923)は、2重化しているプロセッサ3200用領域4480からディスク装置7000へ格納する(ステップ924)ことによりデータ損失を防止することができる。

【0035】ホストコンピュータ1000からの読み込み要求時は、プロセッサ3200は、上記書き込み処理同様、自プロセッサ処理担当論理ボリューム(LUN)への処理であることを認識(ステップ902)した後、処理種別の判定を行う(ステップ903)。I/O処理が読み込み処理であることを認識すると、データ転送制御部3300とDRV I/F制御部3500によりデータをディスク装置群7000からキャッシュ3400のコントローラ3000用領域3480に格納し(ステップ906)、ホストコンピュータに転送する(ステップ907)。

【0036】次にホストコンピュータ1000からコントローラ4000担当論理ボリュームへのI/O処理について説明する。

【0037】ホストコンピュータ1000からの書き込み要求時、プロセッサ3200は、まず、共通メモリ領域3410内の論理ボリューム担当プロセッサ情報3420によって、処理要求論理ボリュームの担当プロセッサ情報を取得し、自処理担当論理ボリュームへの処理かの判定を行い(ステップ902)、処理担当外論理ボリュームへの処理であることを認識する。次に、処理種別の判定を行い(ステップ908)、書き込み処理であることを認識する。そして、ホストコンピュータ1000からの書き込み論理データをキャッシュメモリのコントローラ3000用領域3480に格納し、書き込み処理をこの論理ボリュームの担当であるコントローラ4000へ要求する(ステップ909)。

【0038】プロセッサ3200は、プロセッサ4200に書き込み処理を要求するために、書き込みデータ論理アドレス、書き込みデータのキャッシュ上の格納アドレス、データ長及び処理種別情報をデータ転送制御部

3300により共通メモリ領域3410、4410内のプロセッサ3200書き込み用メモリ内のプロセッサ4200への要求用エリアに2重に格納する。ここで、処理種別情報とは、書き込み処理か読み込み処理かを判断する情報である。プロセッサ4200は、例えば10msといった一定時間で、共通メモリ領域3410、4410の自プロセッサへの要求用エリアを参照にいき、他プロセッサからの要求を認識する。

【0039】図6は、プロセッサ3200からの処理要求を受信したときのプロセッサ4200の処理を示すフローチャートである。前述の方法により、プロセッサ3200からの要求を認識(ステップ931)したプロセッサ4200は、プロセッサ3200書き込み用メモリ内のプロセッサ4200への要求用エリア内の処理種別を参照し、書き込み処理要求であることを認識する(ステップ932)。そして、プロセッサ4200は、プロセッサ3200書き込み用メモリ内のプロセッサ4200への要求用エリア内の書き込み論理アドレス、書き込みデータのキャッシュ上の格納アドレス、データ長を取得し(ステップ933)、キャッシュ3400内の該格納アドレスからデータ長分の書き込みデータをプロセッサ4200用領域3490と4490に、その管理情報である書き込み論理アドレスとデータ長と共に、2重に格納する(ステップ934)。そして、終了情報を共通メモリ領域3410、4410内のプロセッサ4200書き込み用メモリ内のプロセッサ3200からの要求に対する応答用エリアに設定することにより、プロセッサ3200に処理終了を通信する(ステップ935)。

【0040】プロセッサ3200は、プロセッサ4200に対する処理要求後は、プロセッサ4200書き込み用メモリ内のプロセッサ3200からの要求に対する応答用エリアを参照することにより、プロセッサ4200の処理の終了を監視(ステップ910)しており(図4参照)、この処理終了の通信を受けて、ホストコンピュータ1000に終了を報告する(ステップ905)。プロセッサ4200は、この後、図5に従ってホストI/O処理とは非同期に、この書き込みデータのディスク装置7000への書き込み処理を行う。

【0041】図4において、ホストコンピュータ1000から読み込み要求があったときは、プロセッサ3200は、上記書き込み要求受領時同様、処理担当外論理ボリューム(LUN)への処理であることを認識した(ステップ902)後、処理種別の判定を行う(ステップ908)。読み込み処理であることを認識すると、プロセッサ3200は読み込み要求論理アドレス、読み込みデータのキャッシュ上の格納許可アドレス、データ長、処理種別情報を共通メモリ領域3410、4410内のプロセッサ3200書き込み用メモリ内のプロセッサ4200への要求用エリアに格納することにより、該LUN処理担当であるプロセッサ4200に読み込み要求を通

信する(ステップ911)。

【0042】図6において、プロセッサ3200からの要求を認識(ステップ931)したプロセッサ4200は、共通メモリ領域内の情報により、読み込み処理であることを認識する(ステップ932)。そして、共通メモリ領域から読み込み要求論理アドレス、読み込みデータのキャッシュ上の格納許可アドレス、データ長を取得する(ステップ936)。次に、データをディスク装置7000からプロセッサ4200用領域4490に格納し、このデータをキャッシュ3400上の格納許可アドレスに格納する(ステップ937)。さらに、共通メモリ領域3410、4410内のプロセッサ4200書き込み用メモリ内のプロセッサ3200からの要求に対する応答用エリアに終了情報を設定することにより、プロセッサ3200に読み込み終了を通信する(ステップ935)。

【0043】図4において、プロセッサ4200の処理終了を監視(ステップ912)していたプロセッサ3200は、この読み込み終了報告を受けて、データをホストコンピュータに転送する(ステップ913)。

【0044】このように、プロセッサ3200は、通常、プロセッサ3200用領域3480と4480を用いて、I/O処理を行う。同様に、プロセッサ4200は、通常、プロセッサ4200用領域3490と4490を用いて、I/O処理を行う。

【0045】このように、プロセッサ毎に使用するキャッシュ領域を固定化することにより、プロセッサ間の排他制御を無くし、プロセッサ台数増加に伴う性能劣化を防ぐことができる。特にホストコンピュータ間でファイル(論理ボリューム)をシェアしないシステムにおいては、接続しているコントローラ内のプロセッサにこの論理ボリュームを割り当てておくことにより、I/O処理のときのプロセッサ間の通信制御を不要とし、さらなる性能向上を可能とする。

【0046】次にコントローラ4000の障害時の自動切り替え/復旧方式について図7、図8を用いて説明する。I/O処理実行中、コントローラ4000の障害を検知したプロセッサは、共通メモリ領域3410を用いて、残りの全プロセッサにコントローラ4000の障害を通信する。この際、コントローラ4000とキャッシュを2重書きしているコントローラ3000内のプロセッサ3200には、処理の引継要求も通信する。本実施例では、プロセッサ3200が障害を検知した場合について説明する。

【0047】図7は、プロセッサ3200がコントローラ4000の障害を検知した場合のプロセッサ3200の処理を示すフローチャートである。プロセッサ3200はI/O処理実行中(ステップ950)、コントローラ4000の障害を検知(ステップ951)すると、前述の方法により、プロセッサ5200、6200にコン

トローラ4000の障害を通信する。そして、障害コントローラをシステムから切り放すため、キャッシュ3400と4400へ2重書きされているホストコンピュータからの書き込みデータ及び共通メモリ領域のデータを、キャッシュ3400への1重書きに変更することをデータ転送制御部3300に指示する(ステップ952)。また、プロセッサ3200からの要求を認識したプロセッサ5200、6200は、共通メモリ領域をキャッシュ3400への1重書きに変更する。次に、プロセッサ3200は、プロセッサ4200の処理を引き継ぐ為に、プロセッサ4200用領域の制御権をプロセッサ3200に切り替える(ステップ953)。これらの処理により、制御権の切り替えが完了し、プロセッサ3200は通常のI/O処理を再開する(ステップ954)。

【0048】図8は、障害が発生したコントローラ4000の復旧処理を示すフローチャートである。コントローラ4000の障害部位が交換(ステップ971)されると、プロセッサ4200は、共通メモリ領域3410を用いて全プロセッサに復旧開始を伝達する(ステップ972)。プロセッサ3200、5200、6200は、この復旧開始の伝達を受けて(ステップ955)、それぞれのコントローラのデータ転送制御部にキャッシュ3400と4400への2重書きを指示すると共に、共通メモリ領域3410、4410を用いて、処理終了の応答をプロセッサ4200に通信する(ステップ956)。この終了報告を全プロセッサから受領(ステップ973)したプロセッサ4200は、キャッシュ4400のデータ回復を行う(ステップ974)。データ回復が完了すると、共通メモリ領域3410、4410を用いて、プロセッサ3200に復旧完了を伝達する(ステップ975)。

【0049】図7において、この完了通知を受けた(ステップ958)プロセッサ3200は、プロセッサ4200用領域の制御権をプロセッサ4200に復旧(ステップ959)させ、共通メモリ領域を用いて、制御権の復旧をプロセッサ4200に伝達する(ステップ960)。図8において、この伝達を受けた(ステップ976)プロセッサ4200は、I/O処理を再開させる(ステップ977)。

【0050】尚、以上の実施例においては、コントローラ毎にプロセッサ、ホストI/F制御部を1つ持った例を示したが、これらの数は任意でも、ホストコンピュータからのコマンドを受け取ったプロセッサが、担当プロセッサに処理要求を伝達することにより、同様に実現できる。

【0051】また、キャッシュの分割方式は、プロセッサ毎に均等ではなく、ユーザの指定により設定/変更可能である。特に、特定プロセッサをホットスタンバイで動作させる場合には、キャッシュ領域をホットスタンバ

イのプロセッサには割り当てないことにより、キャッシュを有効に利用することができる。又、プロセッサの負荷に応じてダイナミックに変更することも可能である。ユーザの指定により分割を行うか、プロセッサの負荷に応じて変更を行うかの指示は、本実施例では、ホストコマンドにより行うが、パネルといった装置を接続し、そこから入力する形を取っても、むろん良い。

【0052】つぎに、コントローラのキャッシュの動的割当の実現方式について、以下、説明する。まず、キャッシュの管理方式について、図9を用いて説明する。

【0053】プロセッサ毎に持つデータ格納エリアは、セグメント983と呼ばれる管理単位に分割されている。セグメントは、セグメント毎にセグメント管理ブロック981（以下SGCBという。）をデータ管理情報内に持ち、セグメントを管理する情報とセグメントアドレスが格納されている。又、これらのSGCBは、そのセグメントの属性によって、ダーティキュー980とクリーンキュー982という2つキューに分けられて接続されている。ダーティキュー980には、ディスク未反映のライトデータを格納しているセグメントのSGCBが接続されており、それ以外のSGCBは、クリーンキュー982に接続されている。

【0054】キャッシュの動的割当を実現するために、プロセッサ毎の負荷情報を共通メモリ領域に持つ。この負荷情報として、例えば、キャッシュ内のクリーンSGCB量を用いる。各プロセッサは、SGCBのクリーン、ダーティ間のキュー遷移契機に、この情報を更新する。プロセッサは、例えば、1分といった一定周期でこの情報を参照にいき、キャッシュを共有しているプロセッサ内で最も負荷の低いプロセッサのクリーンキューから最も負荷の高いプロセッサのクリーンキューへ、その負荷が同じになるまでSGCBと管理セグメントを移行させる。この際、使用中のSGCBは、移行対象外とする。移行の際は、SGCBの格納データ情報はクリアする。この移行の間は、プロセッサ通信を用いて、移行を行うプロセッサのI/O処理はとめる。

【0055】また、以上の実施例においては、2台のコントローラ間でキャッシュを共有し、各々、対コントローラのキャッシュに2重書きする例を示したが、キャッシュ領域がプロセッサ毎に分割されていれば、そのキャッシュの共有化方式、多重書き方式は、任意の方式でも、同様に実現できる。

【0056】キャッシュ多重書きの例を図10に示す。

(1)は、装置全体でキャッシュを共有しあい、2重書きする方式である。つまり、プロセッサ3200はキャッシュ3400、4400を用いて、プロセッサ4200はキャッシュ4400、5400を用いて、プロセッサ5200はキャッシュ5400、6400を用いて、プロセッサ6200はキャッシュ6400、3400を用いて2重書きを行っている。

【0057】(2)は、装置全体でキャッシュを共有しあい、全キャッシュに多重書きする方式である。つまり、プロセッサ3200、4200、5200、6200は、それぞれキャッシュ3400、4400、5400、6400を用いて、多重書きを行っている。このケースにおいて、コントローラが障害となった場合は、キャッシュを共有しているプロセッサ間でもっとも負荷の低いプロセッサが、障害コントローラ担当論理ボリュームの処理を引き継ぐ。これらのケースにおいては、任意のプロセッサが障害コントローラ担当論理ボリュームの処理を引き継げるように、ディスク側のデータバスを、装置内の全ディスク装置、全コントローラで共通のバスに接続しておく。もちろん、これらの多重書き方を装置内で混在させることも可能である。これらの多重書き方式の指定は、共通メモリ領域3410、4410に多重書き情報を持ち、各々のプロセッサ3200、4200、5200、6200が、この情報を元に、書き込みデータの転送方式をデータ転送制御部3300、4300、5300、6300に指示することにより実現できる。

【0058】

【発明の効果】本発明によれば、コントローラ及びキャッシュメモリを2重化した記憶サブシステムにおいて、各コントローラにキャッシュメモリの一部及び論理ボリュームを割り当てることによりキャッシュメモリに対するコントローラ内のプロセッサ間の排他制御が無くなるため、複数プロセッサ化による応答性能劣化を防ぐことができる。

【0059】また、複数のキャッシュへ多重書きすることにより、キャッシュ障害時には、多重書きしている他キャッシュからディスクに書き込むことができるため、データロストを防ぐことができる。さらに、コントローラ障害時にキャッシュメモリの制御を正常なコントローラに切り替える手段とコントローラ障害から復旧する手段を設けることにより、システムを無停止で運用することができる。

【図面の簡単な説明】

【図1】本発明の概要を表す構成図である。

【図2】本発明の実施例である制御装置の構成図である。

【図3】本発明の実施例であるコントローラのキャッシュの構成を示す図である。

【図4】本発明の実施例によるコントローラのホストからのI/O処理の動作を示すフローチャートである。

【図5】本発明の実施例によるコントローラのキャッシュ内のデータをディスク装置に格納する動作を示すフローチャートである。

【図6】本発明の実施例による他のコントローラから処理要求を受けとったコントローラの制御装置の動作を示すフローチャートである。

【図 7】本発明の実施例による他のコントローラの障害を検出したコントローラの動作を示すフローチャートである。

【図 8】本発明の実施例による障害が発生したコントローラの復旧処理の動作を示すフローチャートである。

【図 9】本発明の実施例によるコントローラにおいて用いられるキャッシュの管理方式を示す図である。

【図 10】本発明の他の実施例二夜コントローラのキャッシュの構成を示す図である。

【符号の説明】

10/11：ホストコンピュータ

20：制御装置

30/40：コントローラ

31/41：コントローラ A 用キャッシュメモリ

32/42：コントローラ B 用キャッシュメモリ

33/43：キャッシュメモリ

50：ディスク装置

1000/1100/1200/1300：ホストコンピュータ

2000：制御装置

3000/4000/5000/6000：コントローラ

3100/4100/5100/6100：ホスト I/F 制御部

3200/4200/5200/6200：マイクロプロセッサ

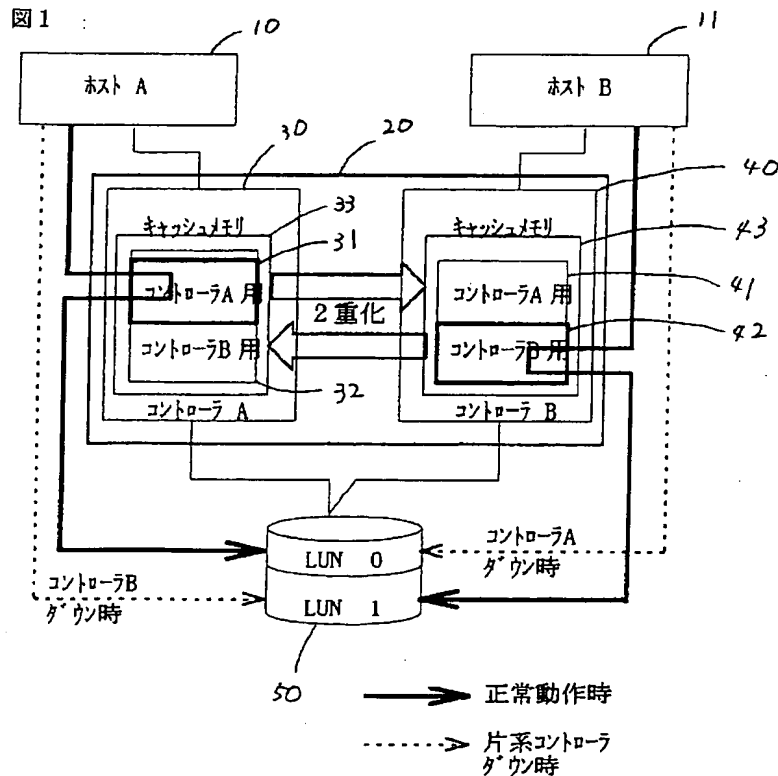
3300/4300/5300/6300：データ転送制御部

3400/4400/5400/6400：キャッシュ

3500/4500/5500/6500：DRVI/F 制御部

7000/7100：ディスク装置群

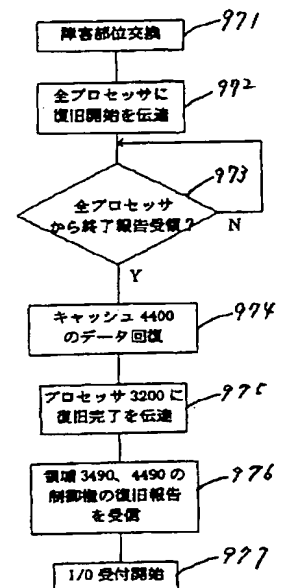
【図 1】



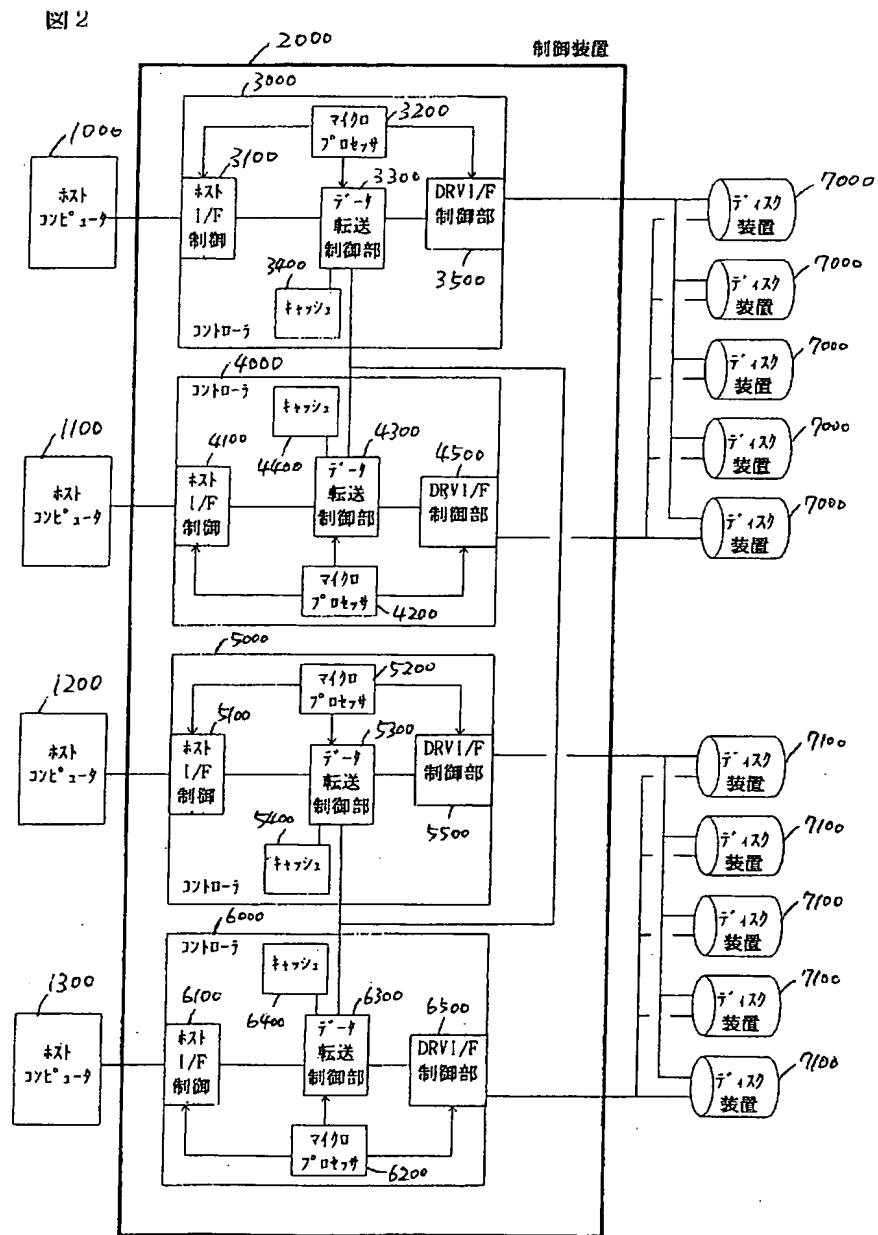
【図 8】

図 8

コントローラ 4000 復旧処理

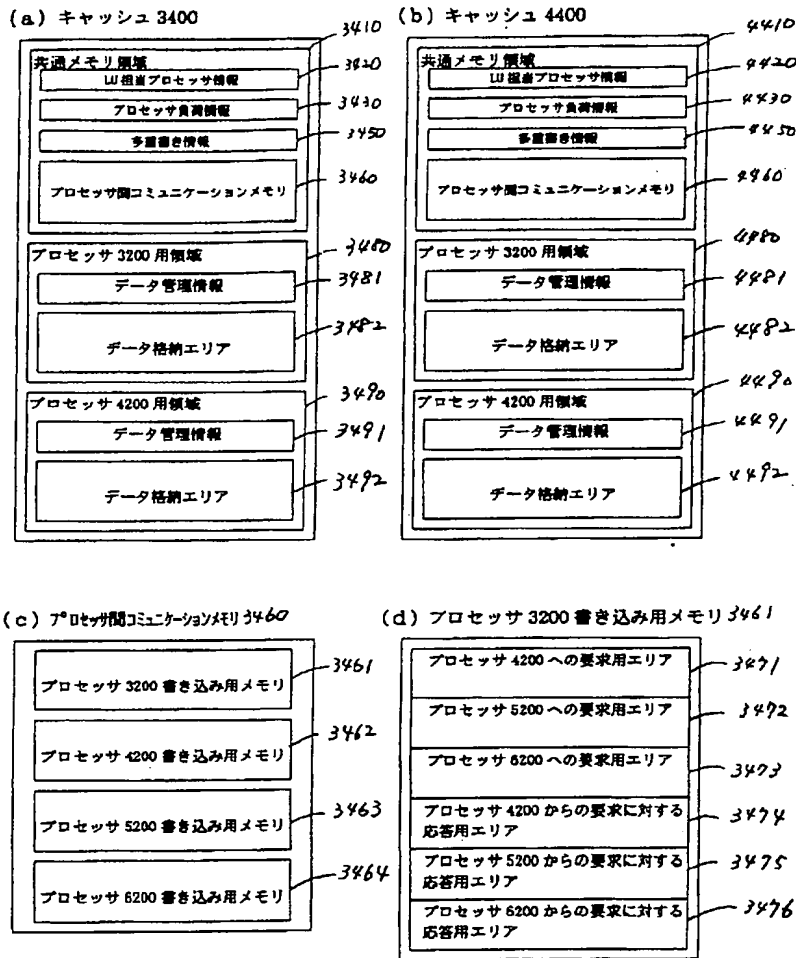


【図2】



【図 3】

図 3



【図 5】

【図 6】

図 5

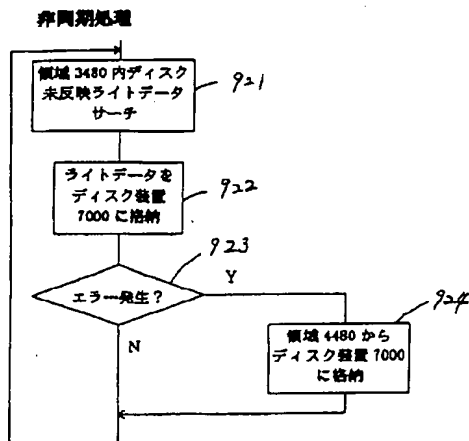
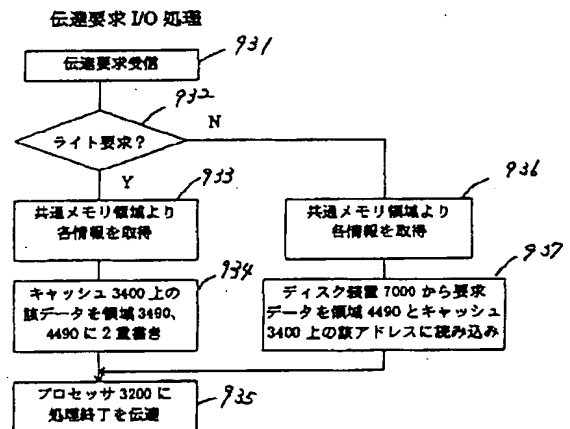
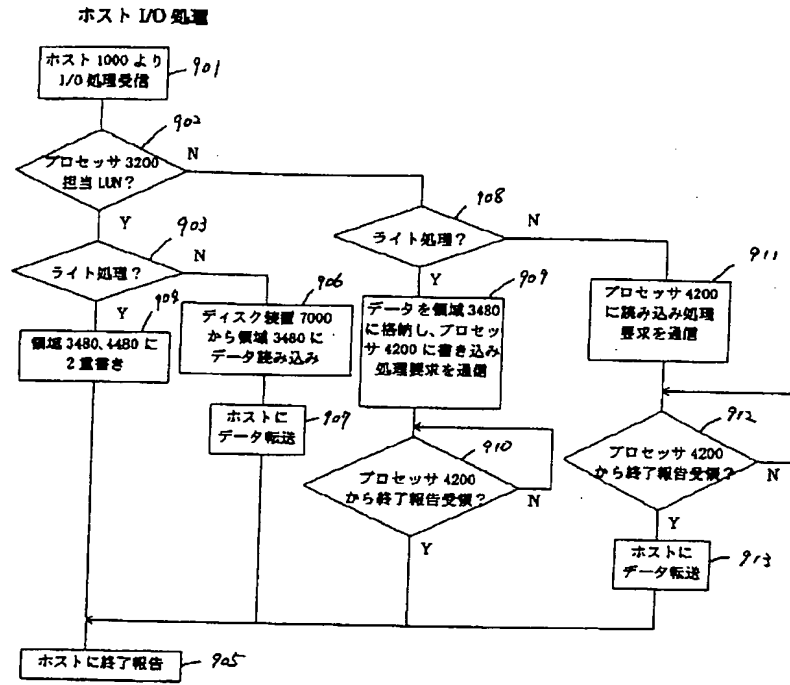


図 6



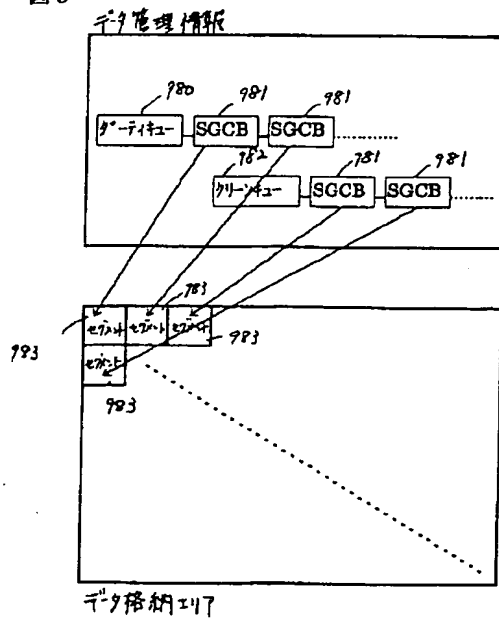
【図4】

図4



【図9】

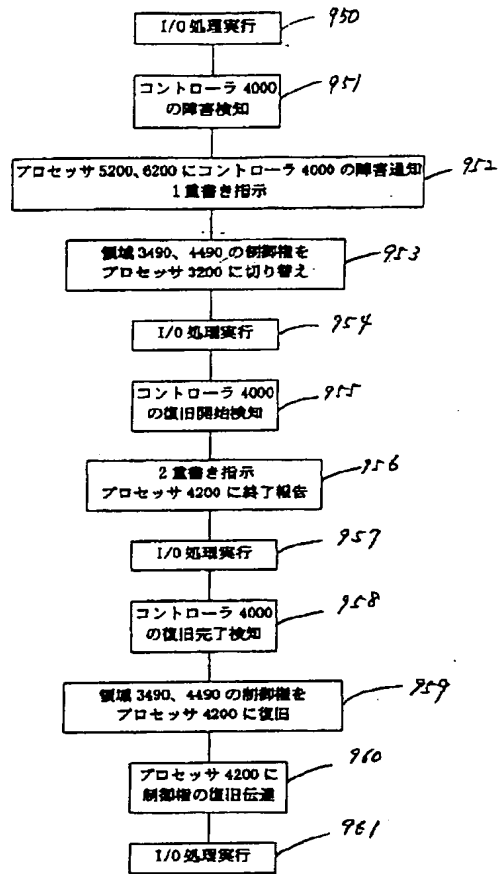
図9



【図 7】

図 7

コントローラ 3000 切り替え/復旧処理



【図10】

図10

